

Absolvování individuální odborné praxe

Individual Professional Practice in the Company

Zadání bakalářské práce

Student: **Veronika Gavenčiaková**
Studijní program: B2647 Informační a komunikační technologie
Studijní obor: 2612R025 Informatika a výpočetní technika
Téma: **Absolvování individuální odborné praxe**
Individual Professional Practice in the Company

Zásady pro vypracování:

1. Student vykoná individuální praxi ve firmě: JetMinds, s.r.o.
2. Struktura závěrečné zprávy:
 - a) Popis odborného zaměření firmy, u které student vykonal odbornou praxi a popis pracovního zařazení studenta.
 - b) Seznam úkolů zadaných studentovi v průběhu odborné praxe s vyjádřením jejich časové náročnosti.
 - c) Zvolený postup řešení zadaných úkolů.
 - d) Teoretické a praktické znalosti a dovednosti získané v průběhu studia uplatněné studentem v průběhu odborné praxe.
 - e) Znalosti či dovednosti scházející studentovi v průběhu odborné praxe.
 - f) Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení.

Seznam doporučené odborné literatury:

Podle pokynů konzultanta, který vede odbornou praxi studenta.

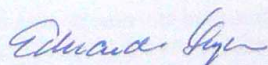
Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Doc. Mgr. Jiří Dvorský, Ph.D.**

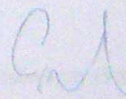
Konzultant bakalářské práce: Ing. Milan Borůvka

Datum zadání: 19.11.2010

Datum odevzdání: 06.05.2011



doc. Dr. Ing. Eduard Sojka
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

„Prehlasujem, že som túto bakalársku prácu vypracovala samostatne. Uviedla som všetky literárne
pramene a publikácie, z ktorých som čerpala.“

V Ostrave 01.05.2011

.....

Na tomto mieste by som rada poďakovala Ing. Milanovi Borůvkovi za pomoc na tejto práci, za jeho prístup a čas, ochotu pomôcť a hlavne za pevné nervy počas našej spolupráce na projekte. Ďalej by som chcela poďakovať Ing. Martinovi Kovářovi, MSc. Sergeovi Cooremanovi, ktorí mi boli obrovskou oporou a taktiež celému kolektívu firmy JetMinds, s.r.o..

Abstrakt:

Táto bakalárska práca sa zaoberá popisom najdôležitejších úloh, na ktorých som pracovala v počas odbornej praxe v rámci vývoja systému Vendavo vo firme JetMinds, s.r.o. Medzi moje najdôležitejšie úlohy patrí úprava systému podľa požiadaviek zákazníka a nasadenie systému, implementácia do prostredia určeného pre testovanie a vytváranie menších dokumentov pre užívateľov.

Kľúčové slová:

Odborná prax, Bakalárska práca, Vendavo, JetMinds, s.r.o.

Abstract:

This bachelor thesis considers description of the most important tasks on which I worked during professional thesis in the scope of development of the Vendavo system in company JetMinds, s.r.o. Among the most important tasks belongs customization according customers requests, deployment and implementation to environment designed to testing and creation of smaller documents for the users.

Keywords:

Professional practice, Bachelor thesis, Vendavo, JetMinds, s.r.o.

Zoznam použitých skratiek a symbolov

AJAX	- Asynchronous JavaScript and XML
API	- Application Programming Interface
B2B	- Business-to-business
CSS	- Cascading Style Sheets
CSV	- Comma-Separated Values
DHTML	- Dynamic HTML
ERP	- Enterprise resource planning
HTML	- HyperText Markup Language
HTTP	- Hypertext Transfer Protocol
IDE	- Integrated Development Environment
J2EE	- Java 2 Enterprise Edition
JDBC	- Java Database Connectivity
JSP	- JavaServer Pages
OLAP	- Online Analytical Processing
OMI	- Object Model Interface
ORM	- Object-relational mapping
SOAP	- Simple Object Access Protocol
SQL	- Structured Query Language
UI	- User Interface
UII	- User Interface Infrastructure
VQL	- Vendavo Query Language
VSL	- Vendavo Scripting Language
XML	- Extensible Markup Language
XSLT	- Extensible Stylesheet Language Transformations

Obsah

Obsah.....	1
Zoznam obrázkov	2
Zoznam tabuliek	3
Zoznam tabuliek	3
1. Úvod.....	4
Profil firiem	5
1.1.1. JetMinds	5
1.1.2. Vendavo.....	5
1.2. Pracovné zaradenie	5
2. Teoretická časť	6
2.1. Aplikácia Vendavo.....	6
2.2. OMI – Object Model Interface.....	6
2.2.1. Čo je OMI?	6
2.2.2. OMI súbory	7
2.3. UII - User Interface Infastructure.....	8
2.4. VSL – Vendavo Scripting Language	9
2.5. VQL – Vendavo Query Language	10
2.6. Pricemart.....	11
2.7. Workflow	11
2.8. Adaptéry.....	12
3. Vendavo Enterprise Pricing Suite	13
3.1. Profit Analyzer.....	13
3.2. Deal Manager	13
3.3. Price Manager	14
3.3.1. Price Console.....	14
3.3.2. Price Program	14
3.3.3. Policy Table.....	14
3.3.4. Price List.....	14
3.3.5. Price Book	14
3.3.6. Workbook	15
3.3.7. Worksheet.....	15
3.3.8. Riadky a stĺpce vo Worksheetoch	15
3.3.9. Fakty	15
3.3.10. Pricing slovník	16
3.3.11. Dimenzia	16
3.3.12. Hierarchia dimenzií	16
3.3.13. Policies	16
3.3.14. Selector.....	16
4. Približný časový prehľad.....	17
5. Niektoré zadané úlohy.....	18
5.1. Workbook šablóny	18
5.2. Stromová štruktúra pre zaradenie workbookov	19
5.3. Kontrola hodnôt v Policy Tabuľkách.....	21
5.4. Úprava Reimbursement Policy Tabuľky.....	22
5.5. Zhodnotenie mojich znalostí a schopností v rámci vyššie uvedených úloh.....	24
6. Záver	26
7. Literatúra	27

Zoznam obrázkov

Obrázok 1: OMI architektúra	6
Obrázok 2: OMI vrstva.....	7
Obrázok 3: UII architektúra.....	8
Obrázok 4: UII framework integrovaný do Vendavo architektúry	9
Obrázok 5: Ukážka workflow mapy.....	11
Obrázok 6: Životný cyklus dokumentu.	11
Obrázok 7: Architektúra adaptérov	12
Obrázok 8: Ukážka z Profit Analyzeru.....	13
Obrázok 9: Ukážka z Deal Manageru (1).....	13
Obrázok 10: : Ukážka z Deal Manageru (2).....	14
Obrázok 11: Ukážka Workbooku	15
Obrázok 12: Ukážka selektoru	16
Obrázok 13: Ukážka Workbook šablóny.....	18
Obrázok 14: Výsledná stromová štruktúra	19
Obrázok 15: Ukážka chybovej správy po validácii	21

Zoznam tabuliek

Tabuľka 1: Približný časový plán.....	17
Tabuľka 2: Ukážka zadania pre 2.01 Target Price Setting Parameters	21
Tabuľka 3: Reimbursement Codes Policy.....	22
Tabuľka 4: Reimbursement Codes to CFN Mappings Policy.....	22

1. Úvod

Ako väčšina študentov, som rovnako aj ja informovaná o tom, že naša fakulta ponúka absolvovanie individuálnej praxe ako alternatívu k bakalárskej práci. Keďže som od leta zamestnaná a táto alternatíva bola pre mňa lákavá začala som zisťovať či je možné nechať si uznať prax aj vo firme, ktorá nie je v zozname firiem schválených našou fakultou. Po tom čo som zistila, že túto možnosť máme, informovala som zamestnávateľa o tejto možnosti a požiadala ho o to aby mi umožnil prax. Moju žiadosť bez problémov prijal a po schválení obomi stranami, firmou i školou mohla moja prax začať. Keďže aplikácia Vendavo je dosť špecifická zo začiatku som bola plná obáv, že môj výber nebol správny a že prax neukončím úspešne. Postupom času som však začala prenikať do tohto systému uisťovať sa v tom, že môj výber bol správny. Bakalárska práca je popisom najdôležitejších úloh, na ktorých som pracovala.

Bakalárska práca má nasledujúcu štruktúru. Prvá kapitola popisuje firmy, pre ktoré pracujem a moje pracovné zaradenie. V nasledujúcich dvoch kapitolách sa snažím oboznámiť čitateľa s danou problematikou, priblížiť jednotlivé prvky, ktoré definujú náš systém a moduly, ktoré obsahuje a o ktorých budem naďalej hovoriť v ďalších kapitolách. V nasledujúcich kapitolách prechádzam od teoretickej stránky k tej praktickej a venujem sa popisu najdôležitejších a časovo náročnejších úloh.

Profil firiem

1.1.1. JetMinds

JetMinds s.r.o. je pomerne mladou firmou založenou v roku 2003, momentálne majúcou 55 zamestnancov. Sídli v Prahe a ďalšie 2 pobočky sa nachádzajú v Hradci Králové a v Ostrave.

Zaoberá sa vývojom a poskytovaním služieb v oblasti podnikového softvéru. JetMinds je strategickým partnerom americkej spoločnosti Vendavo v Európe už od roku 2007, s ktorou spolupracuje na projektoch implementácie pricingových riešení napríklad v spoločnostiach Siemens, Sony Ericson, Danisco, MAN, Medtronic, Michelin, Philips a mnohých ďalších.

Konzultanti sa podieľajú nielen na úpravách softvéru pre konkrétneho zákazníka, ale aj na vývoji tohto riešenia priamo v spoločnosti Vendavo v USA. Medzi ďalšie činnosti patrí podpora koncovým zákazníkom, testovanie kvality a realizovanie pridanej hodnoty u zákazníkov.

1.1.2. Vendavo

Vendavo je popredným svetovým poskytovateľom v oblasti price managementu a optimalizácie pre B2B spoločnosti po celom svete, ktoré pôsobia napríklad v oblasti high-tech odvetvia, chemického priemyslu, priemyselnej výroby, distribúcie, spotrebných výrobkov.

SAP, ktorý je popredným svetovým poskytovateľom podnikového softvéru, majúci viac ako 33,200 zákazníkov vo viac ako 120 krajinách je významným strategickým partnerom Vendava a využíva jeho komplexného riešenia správy cien a ziskov. Je predávaný po celom svete pod názvom SAP Price and Margin Management.

Prostredníctvom Vendavo Enterprise Pricing Suite je možné kontrolovať celý proces „cenotvorby“ od segmentácie až po uzavretie obchodu. Všetky tieto zlepšenia stávajúcich procesov dokážu priniesť významné zvýšenie zisku, obvykle medzi 10% až 30%.

1.2. Pracovné zaradenie

Vo firme JetMinds s.r.o. momentálne pracujem na pozícii „Technický konzultant pre software Vendavo“. Na pozíciu konzultanta bol zvolený Milan Borůvka, ktorý tu pracuje na pozícii Solution Architect / Lead Configuration Engineer, a s ktorým spolupracujeme na vývoji softvéru pre spoločnosť Medtronic.

Náplňou mojej práce je:

- Spolupráca na vývoji Vendavo systému pre európskych zákazníkov
- Úprava aplikácie podľa požiadaviek zákazníka
- Nasadenie systému a implementácia do IT prostredia klienta
- Tvorba technickej dokumentácie
- Tvorba automatických textov
- Spolupráca na časových odhadoch
- Komunikácia s architektom a biznis konzultantmi v angličtine

V praxi uplatňujem skúsenosti s týmito technológiami:

- Programovací jazyk Java, aplikačný framework J2EE
- Administrácia aplikačného serveru JBoss
- Jazyky XML a SQL
- Znalosť databázového systému Oracle
- Práca s vývojovým prostredím Eclipse IDE

2. Teoretická časť

2.1. Aplikácia Vendavo

Vendavo ako aplikácia má viacero modulov cez analytický modul (OLAP s in-memory databázou), modul pre správu pricing policies, generovanie price listov, definíciu workflow, tvorbu zmlúv, integráciu s ERP systémami primárne SAP ERP.

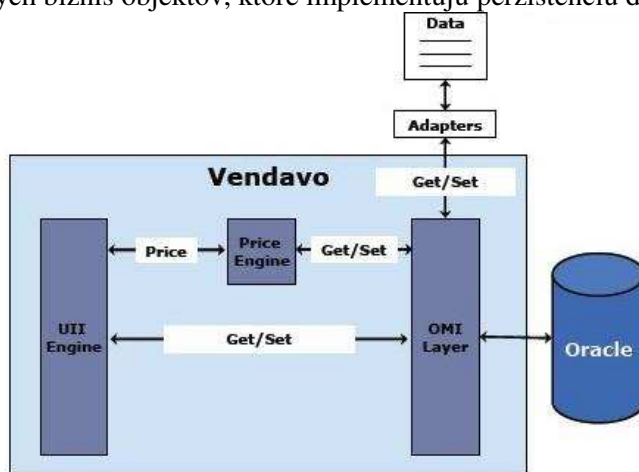
Pricing logika je napísaná v skriptovacom doménovo špecifickom jazyku. Základom je Java, vlastné ORM, AJAXový UI framework, analytické grafy vo Flexe, Web Service integračná vrstva. Vendavo beží nad Oracle a DB2 databázami v prostredí aplikačných serverov JBoss, WebSphere, Weblogic a SAP NetWeaver.

Vendavo taktiež podporuje out-of-box integráciu so SAPom prostredníctvom SOAP.

2.2. OMI – Object Model Interface

2.2.1. Čo je OMI?

OMI je XML objekty popisujúci jazyk používaný pre generovanie Java tried, databázových tabuliek a adaptérov. Taktiež je to architektonická vrstva. OMI súbory sú definície dát, ktoré obsahujú inštrukcie pre vytvorenie alebo modifikáciu dátovej štruktúry pre podnikové koncepty ako zákazníci, produkty, užívatelia a podobne. OMI jazyk poskytuje možnosť modifikovať biznis entity pridaním nových polí alebo modifikáciou už existujúcich polí prostredníctvom XML súborov. OMI entity definované v XML súbore sú automaticky prekladané do Java objektov a do stĺpcov a tabuliek v príslušnej databáze. OMI vrstva automatizuje všetku interakciu s databázou. Mapovanie medzi Javou a databázovými tabuľkami a stĺpcami je udržiavané automaticky. Keď pridáte entitu a pole do aplikácie, OMI vrstva, v spojení s build procesom, automaticky modifikuje príslušnú databázu a automaticky asocjuje nové tabuľky a stĺpce s príslušnými Java objektmi. OMI umožňuje využiť a rozšíriť rozsiahlu knižnicu preddefinovaných biznis objektov, ktoré implementujú perzistenciu do databáze.

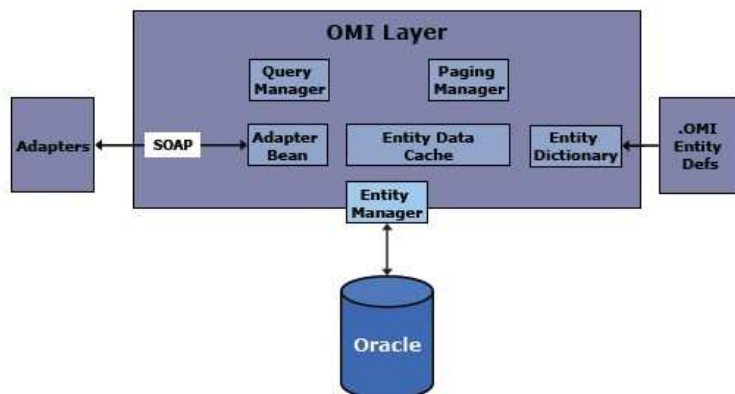


Obrázok 1: OMI architektúra

OMI vrstva je dátová vrstva pre Vendavo. Je zodpovedná za riadenie všetkých dát spracovávaných Vendavom. Konkrétne je zodpovedná za:

- Predkladanie dát Vendavo databáze
- Dotazy na Vendavo databázu
- Cachovanie dát v pamäti
- Sprostredkovanie dát pre UI engine keď je o ne zažiadané
- Prijatie nových dát z UI engine a ich uloženie
- Prijatie nových dát prostredníctvom adaptéru a ich uloženie
- Sprostredkovanie dát adaptéru, keď je o ne zažiadané

OMI vrstva je služba, ktorá oddeľuje zvyšok Vendavo aplikácie od správy dát. OMI vrstva optimalizuje načítanie dát, tým že často používané dáta a dotazy uloží v pamäti. Kód ako adaptéry alebo UI engine nepotrebujú vedieť, či dáta ktoré žiadajú sú načítavané z pamäte alebo disku.



Obrázok 2: OMI vrstva

OMI vrstva uplatňuje objektovo relačné mapovanie a poskytuje všetky služby súvisiace s dátami zvyšku aplikácie.

To zahŕňa:

- Entity Manager zodpovedný za perzistenciu. Entity Manager načíta dáta z databáze a generuje inštancie entít. Keď musia byť dáta uložené Entity Manager dekonštruje inštanciu a uloží jej dáta do tabuliek.
- Entity Data Cache udržiava dáta v pamäti. Keď je zažiadané o objekt, OMI vrstva ho skúsi dostať najskôr z cache.
- Paging Manager spravuje stránkované dáta pre UI. Napr. keď UI zobrazuje zoznam zákazníkov, iba prvých 10 alebo 30 môže byť poslaných klientovi, kým zvyšok je spravovaný Paging Managerom a odosielaný keď je to potrebné.
- Query Manager je zodpovedný za dotazy na objekty.
- Adapter Bean cez ktorý adaptéry importujú a exportujú dáta.
- Entity Dictionary spravuje všetky Entity Definitions pre bežiacu inštanciu. Tieto sú analyzované keď štartuje Vendavo a Entity Dictionary je vytvorený, definujúci kompletnú štruktúru Vendavo entít. Keď sú dáta potrebné, Entity Manager využije Entity slovník aby zistil akú štruktúru každej entity zostaviť a ako entita mapuje na tabuľky v databáze.

2.2.2. OMI súbory

OMI súbory sú definície dátových štruktúr potrebných pre OMI vrstvu. Neplet' si OMI súbory so samotou OMI vrstvou. OMI súbory definujú entity - sú pasívne a samé o sebe nemôžu spraviť nič. OMI vrstva je softvérový komponent, ktorý môže čítať OMI súbory a spravovať entity.

Príklad OMI súboru:

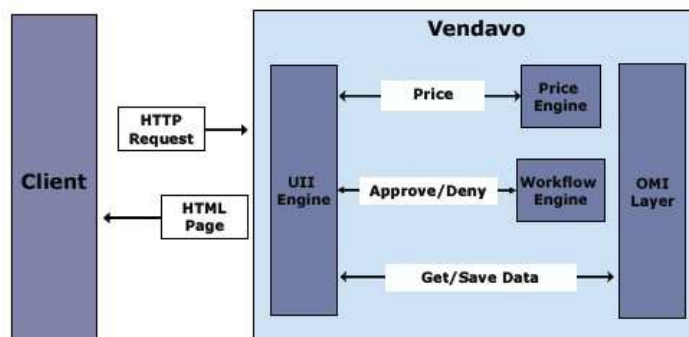
```

<OMI>
  <entity-extension name="com.vendavo.util.api.VCountry">
    <field label="Area" name="Area" type="Area"/>
    <field label="SalesOrgs" name="SalesOrgs" type="VSalesOrg" vector="true"/>
    <field-list name="SeedData">
      <field-ref name="VName"/>
      <field-ref name="VLabel"/>
      <field-ref name="Area"/>
    </field-list>
    <field-list name="SeedData_Country_SalesOrg">
      <field-ref name="VName"/>
      <field-ref name="SalesOrgs"/>
    </field-list>
  </entity-extension>
</OMI>

```

2.3. UII - User Interface Infastructure

User Interaction Architecture



Obrázok 3: UII architektúra

Vendavo je založené na variácii štandardnej J2EE, webovo založenej architektúre aplikácie. Prehliadač klienta zaháji konverzáciu vygenerovaním HTTP požiadavky. Vendavo server odpovie vrátením HTML formuláru. Formulár je vytvorený z jednej alebo viacerých DHTML stránok, ktoré vkladajú JavaScript komponenty aby umožnili užívateľovi interakciu so stránkou (napr. drop-down, tlačidlá, polia pre vkladanie textu atď.)

Nakoniec bude užívateľ komunikovať s kontrolným prvkom, čo spôsobí že prehliadač vygeneruje novú požiadavku pre server. Server odpovie s novým HTML formulárom a tak ďalej, kým sa užívateľ neodhlási alebo neskončí jeho session z dôvodu nečinnosti.

Komponent Vendavo aplikácie, ktorá generuje tieto HTML formuláre sa nazýva UII engine.

Je zodpovedný za nasledujúce 2 veci:

- Vygenerovanie HTML formuláru
- Spracovanie HTTP žiadosti

Za UII engine sú 3 komponenty, ktoré podporujú spracovanie HTTP žiadosti. Najdôležitejšia sa nazýva OMI vrstva. Tá je zodpovedná za dáta. UII engine k nej musí pristupovať vždy keď je potrebné získať alebo uložiť dáta. Ostatné komponenty sú Price Engine a Workflow Engine. Tieto spúšťajú „logiku aplikácie“. UII ich necháva práce vykonávajúce logiku aplikácie a kalkulovalie nových hodnôt dát.

UII poskytuje framework pre konfiguráciu a generovanie Vendavo stránok (užívateľského rozhrania).

Poskytuje:

- Silný, flexibilný, na XML založený komponenty definujúci jazyk pre design-time konfiguráciu a prispôbenie
- Rýchly UII engine, ktorý dynamicky generuje HTML stránky aplikácie, ktoré užívateľ môže vidieť v prehliadači v run-time
- Handlers pre interakciu s Vendavo biznis objektmi

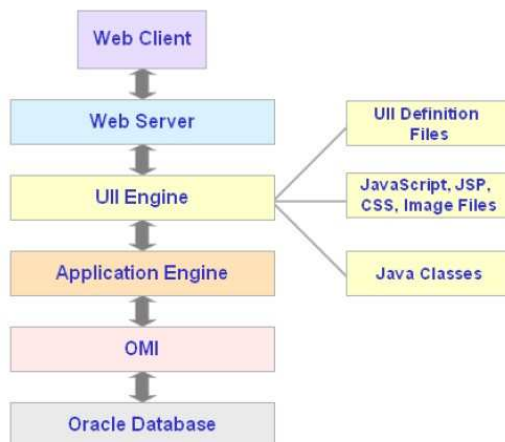
Tak ako OMI prichádza so skupinou preddefinovaných, integrovaných konfiguračných súborov, ktoré je možné použiť ako štartovací bod pre implementáciu.

UII prichádza v 2 úrovniach:

- Základná: poskytuje štandardné UII komponenty, ako šablóny, widgety, funkcie, jazykové konštrukcie a ostatné funkcie, ktoré umožňujú vývojárovi spraviť väčšinu stránok pre zákazníkovu implementáciu.
- Rozšírená: poskytuje komplexnú UII funkcionalitu pre definovanie vlastných UII komponent pre špeciálne účely nepokryté v základnom UII.

V podstate základné UII poskytuje komponenty, ktoré je možno použiť alebo rozšíriť. Rozšírené poskytuje všetky nástroje pre definovanie vlastných komponentov.

Nasledujúci obrázok znázorňuje ako je UII framework integrovaný do Vendavo architektúry:



Obrázok 4: UII framework integrovaný do Vendavo architektúry

UII framework zahŕňa tieto komponenty:

- UI Engine: Run-time procesor, ktorý spracováva požiadavky Web klienta a ako odpoveď generuje stránky, ktoré sú streamované naspäť Web klientovi.
- UII súbory: Obsahujú komponenty definujúci jazyk založený na XML, ktorý umožňuje konfigurovať Vendavo UII.
- Java triedy: Obsahu Java implementácie funkcií, widgetov a ostatných UII komponent.
- Podporované súbory: JavaScript implementácie (ako funkcie), JSPs, CSS súbory, obrázky

Ukážka UII kódu pre drop-down:

```

<DisplayField
  idRef = "SalesOrg"
  isEditable = "true"
  width = "200">
  <view.field.VEntity
    showChooser = "true"
    filterExpr = "$VSL{PriceSettingFuctions.getSalesOrg(entity)}">
  </view.field.VEntity>
</DisplayField>
  
```

2.4. VSL – Vendavo Scripting Language

VSL je skriptovací jazyk, ktorý podporuje pricing a s workflow spojené výpočty.

VSL je používané k:

- Výpočtu hodnôt na formulároch
- Validáciu hodnôt na formulároch

VSL je schopné:

- Načítať z polí na formulároch
- Načítať z polí ostatných entít prostredníctvom bodkovej notácie
- Prehľadať všetky entity aby našlo a prečítalo dáta
- Nastaviť hodnoty polí na formulároch

VSL je zvyčajne používané v nasledujúcich situáciách:

- Výpočet hodnoty samotnej: VSL počíta hodnoty pre jedno alebo viacero polí. S poľami sa zaobchádza ako s premennými. VSL vyzerá ako jednoduchý matematický výpočtový kód

- Vodopádový výpočet: pre jedno pole, VSL nepočíta iba hodnotu poľa, ale taktiež celý vodopád smerom hore k finálnej hodnote poľa.
- Validácia: VSL hodnotí polia na forme k určeniu či/ako nastane vodopád po tom, čo užívateľ klikne na Predložiť.

Ukážka jednoduchšej funkcie pre zmenu meny podľa organizácie:

```
function setCurrentency{
    parameter entity { description ""; }
    vsl{
        if not IsNull(entity.SalesOrg) then
            currencyCode = entity.SalesOrg.Currency.VISOCODE;
            thePrice = entity.ReimbursementPrice;
            let entity.ReimbursementPrice = NewMoney(thePrice,currencyCode);
            return entity;
        end;
    }
}
```

2.5. VQL – Vendavo Query Language

VQL na SQL založený jazyk slúži na dotazovanie dátových objektov. VQL je využívané vo viacerých oblastiach systému, vrátane generovania reportov, užívateľského rozhrania a importu/exportu dát.

Podľa vzoru známeho SQL SELECT príkazu, VQL uľahčuje pracovať programovo s dátami v OMI entitách, umožňuje definovať filtre, ktoré načítajú vybranú podmnožinu entít pre daný typ entity. VQL podporuje väčšinu základných operácií pre dotazy, vrátane implicitných joinov.

VQL podporuje dotazy ako proti OMI tak proti Pricemartu.

Zovšeobecnená základná forma

Nasledujúca štruktúra znázorňuje základnú formu použiteľnú ako pre OMI tak Pricemart VQL.

```
SELECT
    <projected_expr>, <projected_expr>, ...
FROM
    <source_expr>, <source_expr>, ...
WHERE
    [<filter_expr>]
GROUP BY
    <group_expr>, <group_expr>, ...
ORDER BY
    <order_expr>, <order_expr>, ...
HAVING
    <filter_expr>
OPTIONS
    (<option_expr>, <option_expr>, ...)
COMPARE
FROM
    <comparsion_source_expr>
WHERE
    [<filter_expr>]
OPTIONS
    (<option_expr>, <option_expr>, ...)
```

Porovnávanie je podporované iba pre Pricemart VQL a povoľuje iba FROM, WHERE a OPTIONS klauzule.

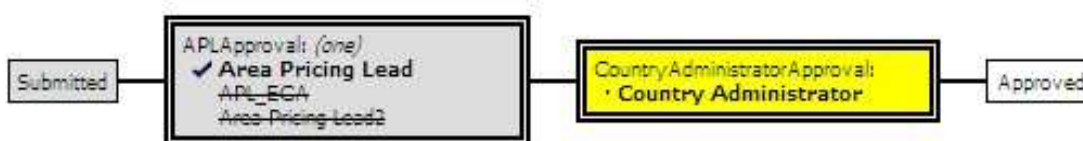
2.6. Pricemart

Pricemart je in-memory databáza, vyvinutá Vendavom pre real-time analýzu cien a zisku obchodov, kvót, kontraktov, objednávok atď.

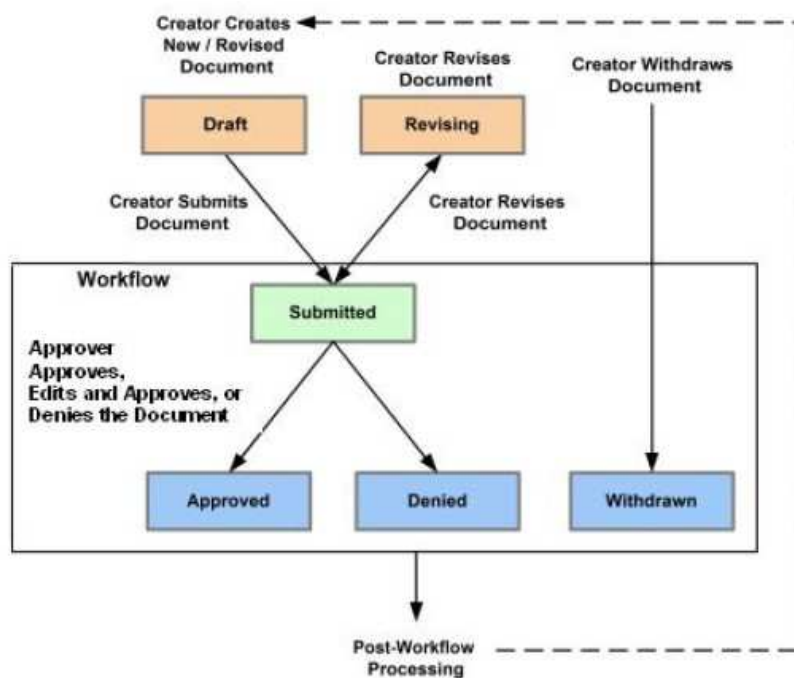
2.7. Workflow

Vendavo poskytuje jednoduchú ale flexibilnú infraštruktúru pre spravovanie procesu schvaľovania pre dokumenty ako ponuky alebo kontrakty, ktoré potrebujú schválenie v rámci organizácie. To znamená, že workflow je proces získania schválenia pre schvaľovanú entitu (ako price list, ponuka, dohoda). Schvaľovaná entita je predložená pre workflow a smerovaná k užívateľom na posúdenie; posudzovateľ to môže odsúhlasiť, zamietnuť alebo editovať.

Keď je vyvolaný schvaľovací workflow, Vendavo vytvorí schvaľovaciu mapu, ktorá znázorňuje, ktorí užívatelia musia entitu schváliť. Vendavo sleduje postup entity cez workflow a aktualizuje mapu vždy keď nejaký posudzovateľ ovplyvní entitu alebo je odoslaná e-mailová notifikácia. Schvaľovateľ môže dokument editovať a schváliť ho. Ak iný schvaľovateľ zakáže entitu, nie je možná žiadna ďalšia akcia, kým ju zakladateľ needituje a znovu nepredloží do workflow procesu.



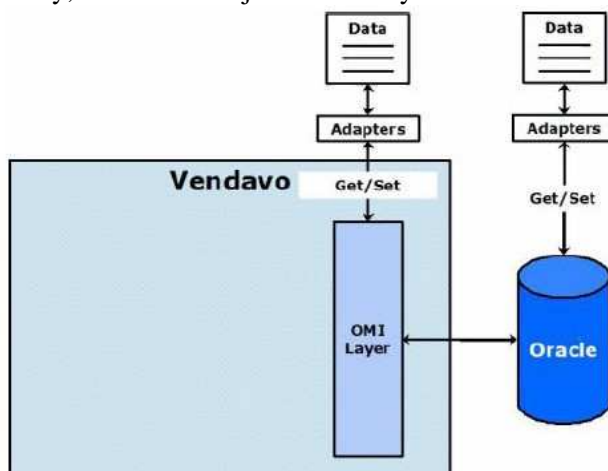
Obrázok 5: Ukážka workflow mapy



Obrázok 6: Životný cyklus dokumentu.

2.8. Adaptéry

Adaptér slúži na načítanie a zápis dát do/zo súboru alebo SOAP správy. Adaptér je kľúčový komponent integračnej architektúry, ktorá umožňuje Vendavu vymieňať si dáta s inými aplikáciami.



Obrázok 7: Architektúra adaptérov

Integračná architektúra zahŕňa:

- Adapter Engine pre čítanie a zápis súborov z/do Vendava. Adapter Engine je Vendavo API pre pohyb dát z alebo do Vendava.
- API v externom systéme, ktoré umožňuje systému čítať/zapisovať textové súbory.
- Prepravný mechanizmus pre kopírovanie jedného súboru z jedného počítača na druhý. Typicky sú integrované 2 systémy bežiacie na rôznych počítačoch.
 - Plánovač
 - (Voliteľne) XSLT prekladač

Typy adaptérov:

- Štandardné Adaptéry je všeobecný termín, ktorý odkazuje na súborovo založené adaptéry pre import alebo export dát pomocou CVS alebo XML súborov. Štandardný adaptér prechádza OMI vrstvou, ktorá poskytuje validáciu. Tento adaptér sa využíva pre malé dátové súbory alebo ak potrebujete komplexnú transformáciu počas načítania dát.
- Adapter Fastpath Import (AFI) sú rýchlejšie ako štandardné a spoliehajú sa na špecifické funkcie Oracle pre zlepšenie výkonu. Neprechádzajú OMI vrstvou a preto nie je prevádzaná žiadna validácia. Je nutné overiť správnosť dát pred tým ako je spustený AFI adaptér. Využívajú sa pre rozsiahlejšie dátové súbory a súbory, ktoré nevyžadujú prílišnú transformáciu.
- JDBC Adaptéry môžu čítať/zapisovať dáta z a do JDBC tabuľky. Poznáme 2 typy JDBC adaptérov. JDBC import adaptéry (dekódovacie) čítajú z JDBC tabuľky konfigurovanej ako zdroj. JDBC export adaptéry (kódovacie) zapisujú do JDBC tabuľky konfigurovanej ako cieľ.

Typy dátových súborov, z/do ktorých môže Vendavo načítavať/zapisovať:

- CSV
- CSV-Vector
- XML

3. Vendavo Enterprise Pricing Suite

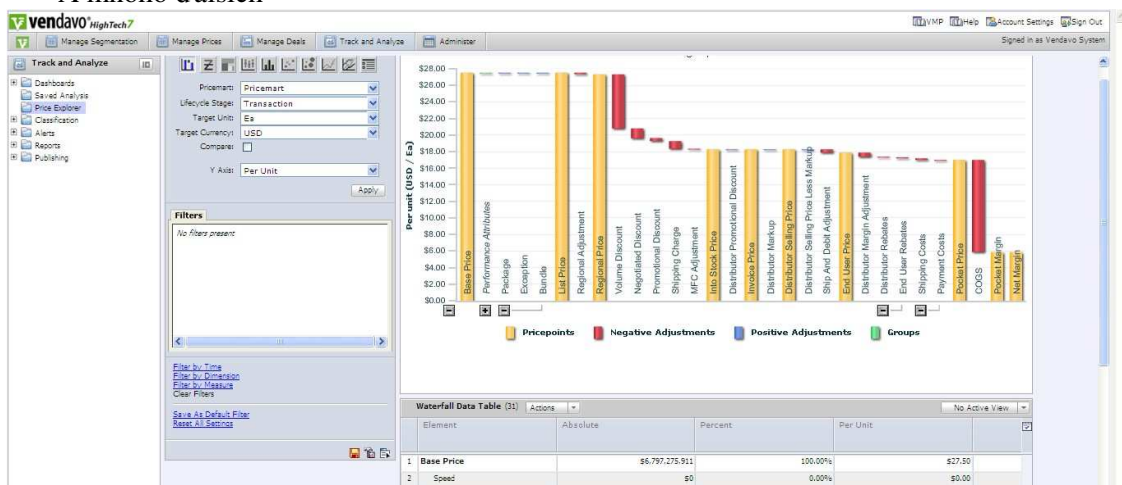
Vendavo Enterprise Pricing Suite je zložený z troch hlavných modulov: Price Manager, kde Vendavo Price Manager nastavuje, kontroluje a spravuje ceny; Deal Manager, kde sú vytvárané a spravované ponuky a Profit Analyzer, kde nastáva analýza cien.

3.1. Profit Analyzer

Vendavo Profit Analyzer pomáha rýchlo identifikovať možnosť zlepšenia marže. Poskytuje podrobný prehľad všetkých komponentov, ktoré definujú jedinečný pricing vodopád pre vašu organizáciu.

Poskytuje možnosti vizualizácie prostredníctvom interaktívnych grafov (price vodopád, scatters, box plots) a bohaté, na roliach založené, dashboards.

- Poskytuje sadu 10 prehľadných grafov s možnosťou rôznych filtrov
- Integrovaný alarm, pokiaľ niektorá z hodnôt prekročí nastavenú medzu
- Dynamické reporty s možnosťou exportu do Excelu
- Klasifikácie a segmentácie produktov
- A mnoho ďalších



Obrázok 8: Ukážka z Profit Analyzeru

3.2. Deal Manager

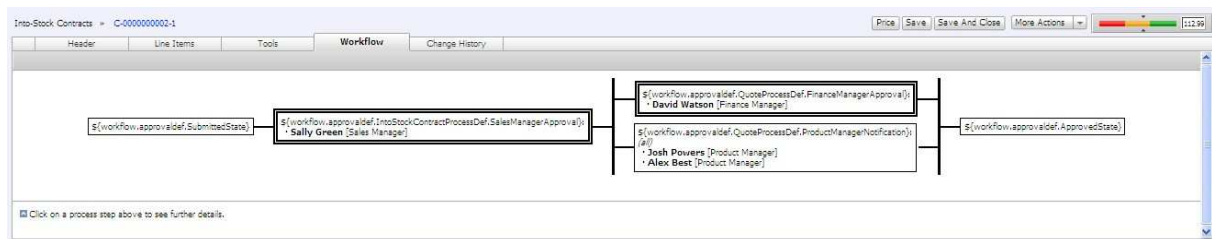
Deal Manager je modul, kde sú poskytované nástroje a informácie, potrebné k vytváraniu, spravovaniu a zjednávaní výhodných obchodov. Poskytuje možné scenáre ziskovosti a umožňuje vybrať najvhodnejšiu možnosť.

Deal Manager je modul, kde sú ponuky, kontrakty a dohody predložené a smerované na schválenie, ktoré je založené na workflow. Ďalej poskytuje možnosť schvaľovania prostredníctvom e-mailu alebo mobilu.

Vo fáze zjednávania dohody obchodní zástupcovia využívajú Deal Manager pre cenové poradenstvo, ktoré je založené na definovaných policiách a pricing programoch.

Line ID	Product	BOM Items	Segment	Suggestion Results	Commitment	Regional Price	Sales Discount	Into Stock Price	Pocket Price	Score	Alert
1	2 NPC10-120A		Unspecified		0.00 Each	\$0.00	0.00%	\$0.00	\$0.00	N/A	
2	4 NPC10-250A		Unspecified		0.00 Each	\$0.00	0.00%	\$0.00	\$0.00	100.00%	
3	3 NPC20-610A		Unspecified		0.00 Each	\$0.00	5.00%	\$0.00	\$0.00	N/A	
4	1 NPC20-620B		Unspecified		0.00 Each	\$0.00	5.00%	\$0.00	\$0.00	33.33%	

Obrázok 9: Ukážka z Deal Manageru (1)



Obrázok 10: : Ukážka z Deal Manageru (2)

3.3. Price Manager

Price Manager umožňuje profesionálny pricing, kontrolovať ceny uvádzané zákazníkom a kontrolovať ako sú s cenami spojené dáta zachytené a reportované. Dáva price analytikovi možnosť kontrolovať pricing obchodov a kontrolovať programy pre zľavy alebo ostatné programy, ktoré majú vplyv na čisté ceny.

Price Manager:

- Dokáže pomôcť s určením optimálnej ceny na základe overených algoritmov
- Umožňuje nastavovať ceny a vytvárať price listy podľa regiónu, času, segmentu, atď.
- Obsahuje rozsiahlu skupinu nástrojov pre podporu správneho nastavenia cien
- Jednoduché schvaľovacie procesy, všade kde sú potrebné

3.3.1. Price Console

Price Console je stavebným kameňom modulu pre nastavovanie cien a administráciu. Je to užívateľské rozhranie, ktoré užívateľovi poskytuje flexibilný spôsob ako organizovať, sledovať a editovať elementy, ktoré už existujú vo Vendavo systéme. Price Console zahŕňa rôzne šablóny, ktoré môžu byť využité ako štartovný bod pre vytváranie a spravovanie cenových programov.

3.3.2. Price Program

Price programy sú kolekciou Dimenzií a policies ktorými sú nastavované ceny alebo poskytované obchodné poradenstvo. Pricing program môže byť aplikovaný na celý biznis alebo na konkrétnu časť biznisu (napr. produkt, segment zákazníkov a/alebo obchodná organizácia). Kľúčový atribút pricing programu leží v jeho schopnosti dediť hodnoty z jednotlivých úrovní hierarchie a potom spravovať výnimky na nižších úrovniach.

3.3.3. Policy Table

Policy tables slúžia na definovanie a spravovanie pricing policies.

3.3.4. Price List

Price listy sú používané pre výpočet cien produktov alebo služieb pre špecifikované časové obdobie. Využívajú policies, ktoré boli nadefinované pre daný produkt a výpočet ceny. Môžu byť definované pre celý biznis alebo podmnožinu biznisov, ako región alebo trh v regióne. Pri ich generovaní užívateľ vyberá množinu dimenzií, ktoré určujú, ktoré produkty majú byť nacenené.

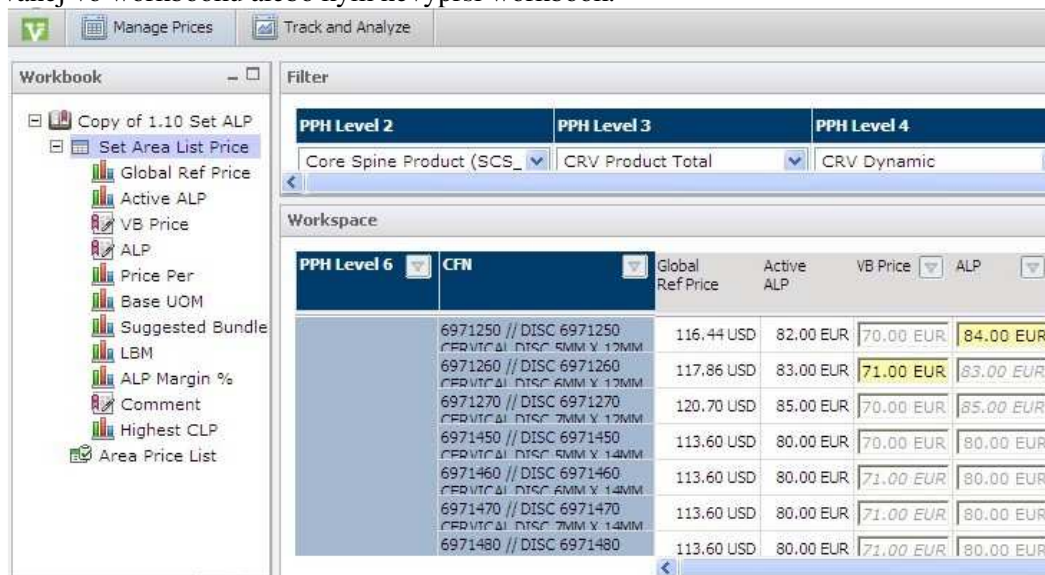
3.3.5. Price Book

V podstate ide o zbierku cien zhromaždených dokopy vo formáte, ktorý môžu biznis ľudia ľahko použiť. Price books obsahujú ceny pre konkrétny kanál alebo obchodný región. Price books môžu obsahovať dodatočné informácie, ktoré zákazník potrebuje. Napríklad môžu obsahovať základné podmienky, informácie o produkte/službe, obrázky, kontakty, informácie a správy pre zákazníkov a tak ďalej. Často tiež obsahujú štandardné zľavy, ako napríklad množstevná zľava. Price books je možné publikovať a distribuovať vo formátoch Excel a PDF.

3.3.6. Workbook

Workbooky vytvárame v Price Console. Slúžia pre nastavovanie cien, riadenie obchodu alebo analýzu pricing dát. Je to kolekcia jedného alebo viacerých worksheetov, ktoré obsahujú pricing policies a analýzy podporujúce policies. Policies vo workbooku sú platné pre určité časové obdobie. Doba platnosti workbooku zodpovedá dobe v biznis kalendári.

Keď je workbook schválený, všetky pricing programy vo workbooku sú schválené. Workbook, ktorý je v stave návrhu obsahuje pricing programy, ktoré nie sú platné. Platnými sa stávajú až po tom čo je workbook schválený. Jeho pricing programy a policies sú potom platné po dobu doby platnosti definovanej vo workbooku alebo kým nevyprší workbook.



PPH Level 6	CFM	Global Ref Price	Active ALP	VB Price	ALP
6971250 // DISC 6971250	CRVITC1A1 D1SC 5MM Y 12MM	116.44 USD	82.00 EUR	70.00 EUR	84.00 EUR
6971260 // DISC 6971260	CRVITC1A1 D1SC 5MM Y 12MM	117.86 USD	83.00 EUR	71.00 EUR	83.00 EUR
6971270 // DISC 6971270	CRVITC1A1 D1SC 7MM Y 12MM	120.70 USD	85.00 EUR	70.00 EUR	85.00 EUR
6971450 // DISC 6971450	CRVITC1A1 D1SC 5MM Y 14MM	113.60 USD	80.00 EUR	70.00 EUR	80.00 EUR
6971460 // DISC 6971460	CRVITC1A1 D1SC 5MM Y 14MM	113.60 USD	80.00 EUR	71.00 EUR	80.00 EUR
6971470 // DISC 6971470	CRVITC1A1 D1SC 7MM Y 14MM	113.60 USD	80.00 EUR	71.00 EUR	80.00 EUR
6971480 // DISC 6971480	CRVITC1A1 D1SC 7MM Y 14MM	113.60 USD	80.00 EUR	71.00 EUR	80.00 EUR

Obrázok 11: Ukážka Workbooku

3.3.7. Worksheet

Worksheetsy sú súčasťou workbookov v Price Console. Tu nastavujeme ceny, definujeme poradenstvo a vykonávame analýzu. Kým worksheet môže obsahovať akúkoľvek kombináciu dát, užitočný worksheet zvyčajne obsahuje iba informácie potrebné k analýze alebo implementácii jedného aspektu pricing programu. Worksheetsy sú vysoko nastaviteľné a sú definované pridávaním faktov a dimenzií.

Typy worksheetov:

- Štandardný worksheet je používaný pre nastavovanie cien a analýzu pricing dát. Štandardný worksheet môže obsahovať fakty, editovateľné policies, kalkulatelné price pointy a kalkulatelné fakty.
- Analytický worksheet obsahuje iba prvky vodopádu, medze a kalkulatelné fakty, Tento worksheet obsahuje iba dáta určené iba na čítanie a nemôže byť používaný pre nastavovanie cien.
- Scoping worksheet je využívaný na nastavovanie scoping pravidiel, pre určenie platných kombinácií a platných hodnôt pre dané časové obdobie. Pricing manažéri ich môžu využiť pre definovanie platnej produktovej konfigurácie pre pricing.

3.3.8. Riadky a stĺpce vo Worksheetoch

Riadky a stĺpce vo worksheetsoch sú ich stavebnými blokmi. Dimenzie sú pridávané ako riadky a stĺpce do worksheetsov. Napríklad, môžete pridať dimenzie ako produkt do riadku a inú dimenziu ako napríklad rezort zákazníka do stĺpca. Priešekník riadkov a stĺpcov poskytuje dáta alebo fakty ako elementy vodopádu alebo policies.

3.3.9. Fakty

Faktom je nazývané políčko worksheetu. Každý fakt reprezentuje odlišnú hodnotu odvodenú z Pricemartu alebo policy table.

Fakty môžu byť organizované do kategórií ako policies či dokonca prvky vodopádu.

3.3.10. Pricing slovník

Pricing slovník obsahuje metadáta využívané v celej Vendavo aplikácii. Zahŕňa najmä nasledujúce dáta:

- Vodopád
- Dimenzie
- Hierarchie dimenzií
- Vzťahy medzi dimenziami
- Priority dimenzií

Definícia vodopádu je základom pre všetky výpočty vykonávané vo VSL. Dimenzie, hierarchie dimenzií a vzťahy medzi dimenziami sú dôležité pre tabuľky politík, pricing programy, workbooks, price listy, price books, cenovú optimalizáciu, podnikovú segmentáciu, atď.

3.3.11. Dimenzia

Dimenzia je vlastnosťou transakcie, ktorá je využívaná pre kategorizáciu dát pre analytické účely. Bežnými dimenziami sú napríklad produkt alebo zákazník.

V Price Console sú dimenzie pridávané do worksheetov. Napríklad ak chceme previesť analýzu nad konkrétnym produktom v danom segmente, vytvoríme analytický worksheet a pridáme dimenzie produkt a segment do daného worksheetu.

3.3.12. Hierarchia dimenzií

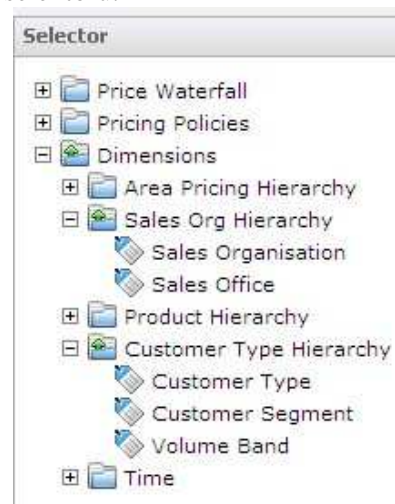
Dimenzie môžu byť spojené hierarchicky. Napríklad hierarchia pre dimenziu produkt môže byť: produktová rodina → produktová línia → produkt. Vzťahy medzi dimenziami robia výber dát efektívnejším. Napríklad v Price Console je možné definovať filtre založené na dimenziách. Keď je dimenzia produktová rodina vybraná ako jeden filter, aplikácia zakáže nelogický výber ostatných ako filtrov.

3.3.13. Policies

Policy umožňuje špecifikovať cenu alebo zľavu spojenú s produktom. Policies sú taktiež využívané k výpočtu cien elementov vodopádu, ktoré sa vyskytujú v grafoch. V Price Console nastavujeme policies ich pridaním do štandardného worksheetu. Sú to jediné editovateľné fakty vo worksheete.

3.3.14. Selector

Selector obsahuje dimenzie, policies a ostatné informácie spojené s pricingom, ktoré sú používané k vytvoreniu worksheetu. Elementy selektora môžu byť pridané do skupín v určenom poradí. Skupiny sa využívajú k vytvoreniu stromu selektora.



Obrázok 12: Ukážka selektora

4. Približný časový prehľad

September		hodín
Práca na workbook šablónach podľa FRD	13.09.2010-20.09.2010	48
Október		
Pridanie price listov do LPS workbookov plus menšie úpravy a opravy vo workbookoch	06.10.2010-07.10.2010	16
Skrytie blokovaných produktov na workbooku	25.10.2010-26.10.2010	16
November		
Úprava policy tabuliek	01.11.2010	8
Definovanie a priradenie oprávnení pre workbooky	05.11.2010	8
Práca na Reimbursement policy tabuľkách	24.11.2010-30.11.2010	40
December		
Práca na Reimbursement policy tabuľkách	01.08.2010-07.12.2010	40
Validácia v policy tabuľkách	09.12.2010-10.12.2010	16
Definovanie a priradenie oprávnení pre policy tabuľky	13.12.2010	8
Február		
Úprava podľa nových požiadaviek: - nové mená workbookov, policy tabuliek a zložiek pre workbooky - nová stromová štruktúra pre zložky - nové názvy faktov	07.02.2011-08.02.2011	24
Vytvorenie nových worksheetov v TAPS workbookoch	09.02.2011-11.02.2011	24
Chybný výpočet CLP - nájsť a opraviť príčinu	14.02.2011	8
Vytvorenie nových worksheetov v TAPS workbookoch	15.02.2011	8
Testovanie: SalesOrg - Currency problém u zákazníka	23.02.2011	8
Marec		
Ďalšie úpravy workbookov podľa nových požiadaviek - odstránenie worksheetov, vytvorenie nových, pridanie nových workbookov	07.03.2011-09.03.2011	24
Zmena policy tabuliek	10.03.2011-11.03.2011	16
Nájdenie a opravenie chyby v Price Analysis workbooku	15.03.2011	8
Testovanie exportu a importu z/do workbooku	17.03.2011	8
Problém s e-mail. notifikáciou - spustenie notifikácie na lokálnej mašine, otestovanie a odhalenie problému	18.03.2011-19.03.2011	16
Testovanie exportu a importu z/do workbooku	22.03.2011	8
Príprava dát pre novú produktovú hierarchiu	24.03.2011	8
Testovanie exportu a importu z/do workbooku	25.03.2011	8
Apríl		
Príprava dokumentácie o nastavení IE pre Vendavo	04.04.2011	8
Otestovanie nastavenie novej prihlasovacej stránky a loga na NetWeaver + pripraviť "how-to" pre zákazníka	11.04.2011	8
Zisťovanie problému s MOTD na servery zákazníka	14.04.2011-15.04.2011	16

Tabuľka 1: Približný časový plán

5. Niektoré zadané úlohy

Počas času stráveného v práci v rámci praxe som riešila úlohy menšieho i väčšieho rozsahu. Vo veľkej miere išlo o takzvaný „bugfixing“, čiže opravovanie chýb v našej implementácii, ktorý spočíval v štúdiu kódu, testovaní a využití rôznych, prevažne interných, nástrojov pre odhalenie a následné opravenie implementácie.

Na tomto mieste by som však rada opísala niektoré zo zložitejších a časovo náročnejších úloh.

5.1. Workbook šablóny

Medzi prvé úlohy, ktoré mi boli pridelené patrilo vytvorenie (prípadne úprava) šablón pre 5 typov workbookov.

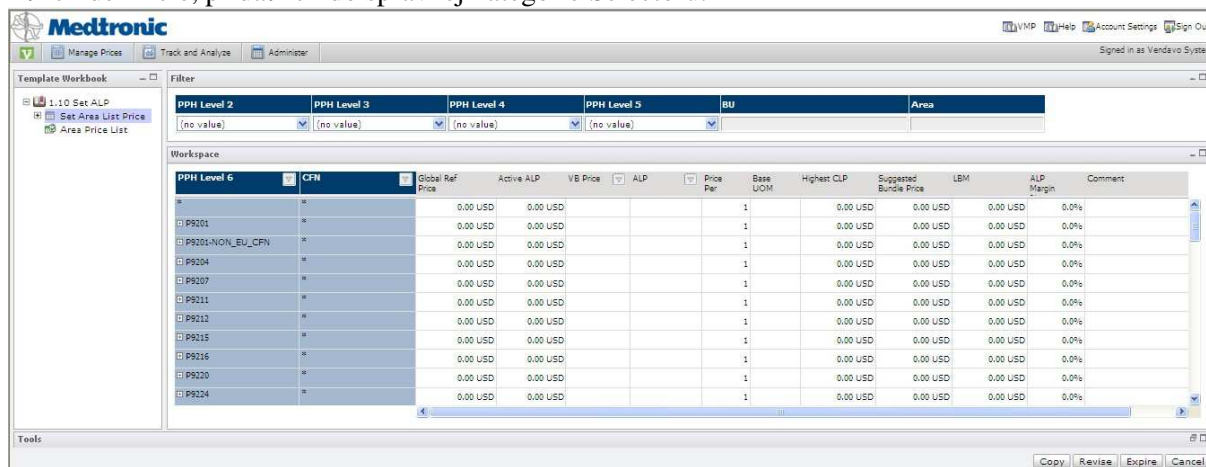
Pre lepšie pochopenie situácie, užívatelia, ktorí využívajú náš systém majú pridelené rôzne role, na základe ktorých môžu administrovať ceny produktov pre určitý región, štát či zákazníka. Pre každú túto úroveň je definované čo všetko môže daný užívateľ vidieť. Či už najvyššia cena nastavená pre produkt v danom segmente, aktuálne nastavená cena, rôzne kalkulované hodnoty, tieto všetky ktoré majú užívateľovi pomôcť nastaviť cenu čo najlepšie. Ďalej je definované aké ďalšie hodnoty spojené s cenou môže daný užívateľ nastaviť. Medzi tieto patria napríklad nejaké prahové hodnoty, pod ktoré by navrhovaná cena nemala klesnúť a podobne. Aby nebolo zakaždým potrebné vytvárať nový workbook úplne od nuly, je možné vytvoriť šablóny, ktoré definujú celú štruktúru workbooku. Z takto pripravenej šablóny užívateľ môže odrazu vytvoriť workbook a pracovať. Mnohí užívatelia nemajú právo vytvárať vlastné workbooks, ale iba vytvoriť workbook zo šablóny. Týmto a ďalšími nastaveniami, ktoré obmedzujú právomoci užívateľa zabezpečíme, že nenastaví hodnoty mimo svoj región a podobne.

Najjednoduchším spôsobom ako vytvoriť takúto šablónu je využitie UI Vendava. Konkrétne Price Console, poskytuje základnú šablónu, do ktorej je možné pridávať worksheets, price listy, filtre, definovať platnosť entity a definovať prvky, ktoré budú jednotlivé worksheets.

Moja úloha spočíva vo vytvorení worksheetu vo workbooku, pridaní správnych filtrov na úroveň workbooku a worksheets. Ďalej pridať do worksheetsov:

- Dimenzie
- Price Setting Forms – pre fakty, ktoré sú dedené alebo dopočítavané vo VSL
- Calculated Facts – vzorec pre výpočet ich hodnoty je možné nadefinovať priamo cez užívateľské rozhranie
- Pricing Programy, ktoré sa využívajú u editovateľných faktov

Prípadne ak niektoré z daných elementov doteraz neboli nikým nadefinované bolo potrebné vytvoriť ich definície, pridať ich do správnej kategórie Selectoru.



Obrázok 13: Ukážka Workbook šablóny

Po vytvorení šablóny je pre jej ďalšie použitie vyexportovať ju pre ďalšie použitie.

K tomu je nutné nadefinovať adaptér pre export a import do systému:

Ukážka adaptéru pre export šablóny do XML a import z XML do systému:

```
<adapter name = "AreaListPriceWorkbookTemplate"
  encoding = "UTF-8"
  exporter = "com.vendavo.platform.adapter.api.XMLEncoder"
  field-list = "ImportExportWithChildren"
  file = "pricesetting/AreaListPriceWorkbookTemplate.xml"
  importer = "com.vendavo.platform.adapter.api.XMLDecoder"
  type = "Workbook">
  <property name = "adapter.export.vql-filter" value =
    "VIsTemplate=true AND VState.VName='approved' AND Type.VName
    = '1.10 ALP Area List Price'"/>
</adapter>
```

Medzi ďalšie úlohy, spojené s workbookmi, patrí vytvorenie „entitlementu“, čo je vlastne definovanie právomocí, jednotlivých užívateľov, vytvorenie stromovej štruktúry pre workbooky, vytvorenie price listov a definovanie jednotlivých funkcií pre výpočet alebo načítanie hodnôt z databázy a v neposlednom rade práca na zmene ich dizajnu a funkčnosti na základe požiadaviek zákazníka, či už zmeny názvov jednotlivých workbookov, prvkov, ktoré obsahujú, zmena filtrov, poradí jednotlivých prvkov pridanie nových alebo naopak odstránenie a podobne.

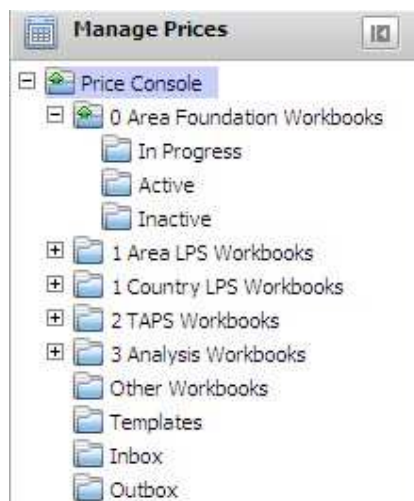
5.2. Stromová štruktúra pre zaradenie workbookov

Ďalšou úlohou bolo pokračovať na štruktúre pre filtrovanie workbookov do jednotlivých zložiek a tým uľahčiť prehľad:

Skrátené zadanie:

- Typ workbooku
 - In progress (draft, submitted, revising, declined)
 - Active (approved and valid today)
 - Inactive (expired or approved and not valid today)

Výsledná štruktúra:



Obrázok 14: Výsledná stromová štruktúra

Funkcia, ktorá definuje filter pre workbooky má cez 150 riadkov a volajú sa v nej ďalšie funkcie, pre zistenie toho, ktoré workbooky môže daný užívateľ vidieť, preto ako ukážku prikladám iba časť kódu pre jeden typ workbooku:

```

vsl {
  let areaPermission = getAreaPermission(user);
  let filterPPH = getFilterBUWorkbook(user);
  let filterAPHArea = getFilterAreaWorkbook(user, areaPermission);
  let filterAPHCountry = getFilterCountryWorkbook(user, areaPermission);
  let filterState = "((VState.VName in ('draft', 'revising',
    'withdrawn', 'denied', 'expired') and VWorkflowOwner.VName =
    ':1') or ':1' in ('SAdmin', 'vendavosystem') or (VState.VName in
    ('approved', 'superceded', 'submitted'))";
  filterState = filterState.replace(":1", user.VName);

  let isAreaLPSWB = "(Type.VName in ('1.10 ALP Area List Price', '1.20
    CLP set by Area', 'Area List Price', 'Country Price Setting for
    APL') and VIsTemplate=false)";

  let query = "(VState.VName = 'approved' ";
  query = CONCAT(query, "and VValidity.to >= today())");
  let filterActive = query;

  let query2 = "(VState.VName = 'approved' and (";
  query2 = CONCAT(query2, " VValidity.to < today())";
  query2 = CONCAT(query2, ")) or (VState.VName = 'expired' ";

  if user.VIsSysAdmin or user.hasRole("SuperAdministrator") then
    let filterInProgress = "(VState.VName in ('draft', 'revising',
      'withdrawn', 'denied', 'submitted'))";
    query2 = CONCAT(query2, "));";
    let filterInactive = query2;
  else
    let filterInProgress = "((VState.VName in ('draft', 'revising',
      'withdrawn', 'denied') and VWorkflowOwner.VName = ':1') or
      (VState.VName in ( 'superceded', 'submitted'))";
    filterInProgress = filterInProgress.replace(":1", user.VName);
    query2 = CONCAT(query2, " and VWorkflowOwner.VName = '");
    query2 = CONCAT(query2, user.VName);
    query2 = CONCAT(query2, "'))";
    let filterInactive = query2;
  end;

  let filterAreaLPSWB = "(" + "(" + filterPPH + " and " + filterAPHArea
    + " and " + isAreaLPSWB + ")" + " or " + isGRP + ")";

  if inProgress then
    filterState = filterInProgress;
  end;
  if active then
    filterState = filterActive;
  end;
  if inactive then
    filterState = filterInactive;
  end;

  if areaLPSWBs then
    if user.VIsSysAdmin or user.hasRole("SuperAdministrator") then
      return filterState + " and " + isAreaLPSWB;
    else
      filter = filterState + " and " + filterAreaLPSWB;
      return filter;
    end;
  end;
}

```

S touto úlohou je taktiež spojená úloha definovať, ktorí užívatelia môžu vidieť ktoré zložky.

5.3. Kontrola hodnôt v Policy Tabuľkách

Pri častej práci s policy tabuľkami som si všimla nutnosti zabezpečiť správnosť odsúhlasených hodnôt, ktoré obsahovali. Keďže doteraz neboli nikde tieto požiadavky špecifikované, po konzultácii s mojím nadriadeným bolo pripraviť zoznam všetkých policy tabuliek a polí, ktoré obsahujú, zo znalosti niektorých požiadaviek a funkčnosti celého systému zostaviť prehľad toho, ktoré polia môžu zostať nevyplnené, ktoré môžu obsahovať nulové, či záporné hodnoty a či dané polia sú alebo nie sú editovateľné a po schválení a prípadných úpravách ďalším nadriadeným sa postarať o následnú implementáciu.

2.01 Target Price Setting Parameters			
Field	Required	Negative value	Editable
International Volume Break Percentile 1	Y	N	Y
International Volume Break Percentile 2	Y	N	Y
Statistical Significance Floor	Y	N	Y
Order Line Types	Y	/	Y

Tabuľka 2: Ukážka zadania pre 2.01 Target Price Setting Parameters

Toto sa dá zabezpečiť nastavením parametrov pre dané polia v OMI a UII súboroch.

To či je nutné dané pole vyplniť definujeme v OMI súbore prostredníctvom parametru „required“:

```
<field label="International Volume Break Percentile 1"
name="Percentile1" type="int" required="true">
```

Či pole môže obsahovať negatívne hodnoty a či je editovateľné nastavujeme prostredníctvom UII. Štandardne je pole editovateľné, takže v tomto prípade sa týmto nemusíme zaoberať a aby pole obsahovalo iba kladné hodnoty zabezpečíme pomocou parametru „nonNegative“

```
<DisplayField idRef="Percentile2">
  <view.field.VNumber nonNegative="true"/>
</DisplayField>
```

Výsledkom zadania nesprávnych hodnôt môže byť napríklad takáto stránka:

International Volume Break Percentile 1: Field cannot be negative.
Statistical Significance Floor: Field cannot be negative.

Policy Record Approval

International Volume Break Percentile 1: Order Line Types:

International Volume Break Percentile 2: Status:

Statistical Significance Floor: Validity: to

Obrázok 15: Ukážka chybovej správy po validácii

5.4. Úprava Reimbursement Policy Tabuľky

Podľa pôvodného návrhu boli kolegom nadefinované 2 policy tabuľky:

Field	Field Type	Value Type	Editable?	Required?	Splice Key?
Reimbursement Code	Policy Table Column	Varchar	Y	Y	Y
Reimbursement Code Description	Policy Table Column	Varchar	Y	Y	Y
Country	Policy Table Column	Text	Y	Y	Y
SAP Sales Org	Policy Table Column	Text	Y	Y	Y
Reimbursement Price	Policy Table Column	Price (set in local country sales org currency)	Y	Y	N
Status	Policy Table Column	Text	Y	Y	N
Valid From	Policy Table Column	Date	Y	Y	N
Valid To Date	Policy Table Column	Date	Y	Y	N

Tabuľka 3: Reimbursement Codes Policy

Field	Field Type	Value Type	Editable?	Required?	Splice Key?
Reimbursement Code	Policy Table Column	Varchar	Y	Y	Y
Reimbursement Code Description	Policy Table Column	Varchar	Y	Y	Y
CFN	Policy Table Column	Text	Y	Y	Y
Status	Policy Table Column	Text	Y	Y	N
Valid From	Policy Table Column	Date	Y	Y	N
Valid To Date	Policy Table Column	Date	Y	Y	N

Tabuľka 4: Reimbursement Codes to CFN Mappings Policy

2. tabuľka obsahovala pole „Reimbursement Code“, do ktorého bolo možné vložiť ľubovoľný reťazec. Čo bola chybná implementácia a mne pripadla na prvý pohľad triviálna úloha zmeniť toto pole tak aby bolo možné vyberať iba z vopred nadefinovaných „Reimbursement Codes“.

Prvotný nápad bol zmeniť toto pole na drop-down, v ktorom by sme mohli vybrať jednu z ponúkaných možností. Problémom u tohto riešenia, ale je nevhodnosť jeho použitia pri väčšom množstve záznamov.

Ďalšou možnosťou ako zobrazíť tieto hodnoty je dialógové okno, ktoré je vlastne pop-up oknom, zobrazujúcim vždy 10 záznamov na 1 stránke, s možnosťou filtrovania.

Pri snahe implementovať dialógové okno sme narazili na problém, že toto okno nie je možné nadefinovať pre polia typu String. Riešením bolo vytvoriť entitu, do ktorej by sme ukladali príslušné hodnoty. Toto mnou bolo úspešne implementované Ale keďže by bolo toto riešenie zbytočným skomplikovaním, prišiel náš Solution Architect s návrhom spojiť tieto tabuľky do jednej a využiť možnosti vektorovo pracovať s produktmi, čo bolo v konečnom dôsledku najoptimálnejším riešením.

Ďalší problém, ktorý bolo potrebné vyriešiť je závislosť medzi jednotlivými poľami. Predajné Organizácie, ktoré môžeme vybrať sú závislé na krajine, ktorú vyberieme. Problém nastáva keď užívateľ najskôr vyberie Predajnú organizáciu a potom krajinu, čím môže nastať situácia, kedy užívateľ zvolí nesprávnu kombináciu dát.

Nasledujúci kus kódu znázorňuje časť definície UI pre Reimbursement policy.

```
<EntityView name="policymanager.PolicyTableRowView"
entityName="com.Medtronic.policymanager.ReimbursementCodesPolicy"
extends="policymanager.VAutoGenTableRowView">

  <Init id = "init">
    <VIf value="{VNot (VOR(VHasRole('SuperAdministrator'),
    loggedinUser.VIsSysAdmin))}">
      <VSet name = "areaPermission" value =
        "$VSL{PriceSettingFuctions.getAreaPermission(loggedinUser)}"/>
      <VSet name = "countryFilter" value =
        "$VSL{PriceSettingFuctions.getFilterCountrySliceConfig(loggedinUser,
        areaPermission)}"/>
      <VSet name="entity" value="{view.VEntity()}" />
    </VIf>
    <VCall function="$VSL{PriceSettingFuctions.setCurrency(entity)}"/>
  </Init>
  ...
  <DisplayField idRef="Country">
    <view.field.VEntity showChooser="true"
      filterExpr="{countryFilter}">
      <VParameterValue name="onChange">
        <controls.VAction actionHandler="policymanager.test">
          <controls.VActionRequestParameter name="entity1"
            value="{view.VEntityId()}" />
        </controls.VAction>
      </VParameterValue>
    </view.field.VEntity>
  </DisplayField>

  <DisplayField idRef="SalesOrg"
    isEditable = "{VAND(VEQ(entity.VState.VName, 'draft'),
    VIsNotNull(entity.Country))}"
    width="200">
    <view.field.VEntity showChooser="true"
      filterExpr="$VSL{PriceSettingFuctions.getSalesOrg(entity)}">
      <VParameterValue name="onChange">
        <controls.VAction actionHandler="policymanager.test2"/>
      </VParameterValue>
    </view.field.VEntity>
  </DisplayField>
  ...
</EntityView>

<ActionHandler name="policymanager.test">
  <RequestParameter name="entity1" type="string"/>
  <action.VSetField entityId="{entity1}" field="SalesOrg" value="" />
</ActionHandler>
<ActionHandler name="policymanager.test2"/>
```

Pri vytvorení novej Reimbursement policy, máme pole Country nastavené ako editovateľné a pole Sales Organization ako needitovateľné. Ako je zreteľné z kódu po zmene obsahu poľa Country sa vyvolá akcia policymanager.test, ktorá vynuluje hodnotu SalesOrg. Po zavolaní tejto akcie sa „refreshne“ celá stránka, čo má za následok prevedenie celého kódu. Čo nám zabezpečí, že teraz už je Sales Organizácia editovateľná, je zavolaná funkcia pre výber platných Sales Organizácií a my si môžeme vybrať jednu z vyfiltrovaných hodnôt.

Definícia funkcie pre výber Sales Organizácií je nasledovná:

```
function getSalesOrg{
  parameter entity { description ""; }
  vsl{
    if not IsNull(entity.Country) then
      let filterIN = "";
      let query = "select salorg.VName from VSalesOrg salorg, VCountry
        cntr where cntr.SalesOrgs.VName = salorg.VName and cntr.VName = :1";
      let qry = VQLQuery(query, entity.Country.VName);
      let iter = qry.getResultSet().iterator();

      while iter.hasNext() do
        let resultrow = iter.next();
        let salorg = resultrow.getValue("salorg.VName");
        if not filterIN.equals("") then
          filterIN = filterIN + ",";
        end;
        let filterIN = filterIN + "'" + salorg + "'";
      end;

      if (iter <> null) then
        iter.close();
      end;
      if (qry <> null) then
        qry.close();
      end;
      let filter = "1=2";

      if filterIN.equals("") then
        return filter;
      end;
      let filter = "VName in ";
      return filter + "(" + filterIN + ")";
    end;
  }
}
```

Ďalej si môžete všimnúť že po zmene Sales Organizácie je vyvolaná ďalšia akcia, ktorá je prázdna a tým pádom jej jedinou úlohou je „refresh“, pri ktorom sa zavolá funkcia setCurrency(), ktorá nastaví menu podľa vybranej Sales Organizácie.

5.5. Zhodnotenie mojich znalostí a schopností v rámci vyššie uvedených úloh

Ako som už spomínala tak práca na workbook šablónach bola jednou z prvých ktoré mi boli pridelené. Preto bola pre mňa najťažšia, ale zároveň aj najprínosnejšia, keďže som sa pri jej plnení začala zoznamovať s princípmi pricingu a prenikať do aplikácie Vendavo. Zistila som akú funkciu majú workbook šablóny, ako ich definovať, ako vytvoriť adaptér, ktorým ich exportujem a tým umožním ich nasadenie do systému a ďalšie využitie.

Pri práci na filtri pre roztriedenie vytvorených workbookov do jednotlivých zložiek som bola „nútená“ preštudovať a pochopiť niektoré funkcie, ktoré boli implementované kolegom, aby som bola schopná pokračovať v jeho práci. Toto bolo prínosné v dvoch smeroch. V prvom rade som sa dozvedela veľa o tom ako efektívne implementovať filtre a zabezpečiť aby užívatelia mohli pracovať iba prvkami systému, ktoré sú preňho na základe požiadaviek prípustné. Ďalším prínosom je naučenie sa čítania cudzieho kódu a jeho pochopenie, čo je dôležité ako pre celkové pochopenie funkčnosti systému, tak prínosné v prípade potreby nahradenia kolegu a pokračovaní v jeho práci.

Pri riešení problému validácie bolo nutné preštudovať iné projekty a systémové kódy pre nájdenie najvhodnejšej implementácie, keďže v tomto mi kolegovia v niektorých prípadoch nevedeli poradiť.

Opäť to bolo prínosné v tom, že človek pochopí ďalšie súvislosti v systéme a naučí sa čítať kód, ktorý nie je písaný jeho rukou. Taktiež to bola úloha, ktorú som prvý krát navrhla sama, takže prínos bol v komunikácii s nadriadeným a zapojenie sa do procesu definovania požiadaviek na systém.

V poslednej úlohe bolo opäť potrebné komunikovať s nadriadeným a spoločnými silami nájsť riešenie, ktoré by bolo najvýhodnejšie, ktoré by najmenej zaťažilo systém, ktoré by vyžadovalo najmenej operácií, skrátka nájsť to najoptimálnejšie riešenie. Pri tejto úlohe som využila vedomostí nadobudnutých v predmetoch zameraných na databáze, konkrétne Teória spracovania dát.

6. Záver

Práca vo firme JetMinds je pre mňa obrovským prínosom. V prvom rade je to obrovská skúsenosť v obore, naviazanie na znalosti zo školy, ich prehĺbenie a pridanie nových. Keďže Vendavo je ako systém dosť špecifický prakticky všetko bolo pre mňa nové a do procesu tvorby systému bolo ťažké sa zapojiť. Obrovskou oporou mi boli vedomosti získané predovšetkým v predmetoch zameraných na programovanie a Teória spracovania dát. Ďalšou skúsenosťou je opustenie školskej lavice a zaradenie sa do tímu, práca na projekte už trochu väčšieho rozmeru, aktívne zapojenie sa do procesu, komunikácia v anglickom jazyku s nadriadenými a kolegami z rôznych krajín. Taktiež je zaujímavé a prínosné zoznámiť sa s problematikou „cenotvorby“. Prax hodnotím veľmi pozitívne, získané znalosti, ktorých je nemálo, v budúcnosti určite využijem, už len z dôvodu, že odovzdaním bakalárky sa moja práca v tejto firme nekončí.

Veronika Gavenčiaková

7. Literatúra

- [1] 7.5_Deal_Manager_User_Guide.pdf, <http://www.vendavo.com/> (bez udania dátumu)
- [2] 7.5_Price_Manager_User_Guide.pdf, <http://www.vendavo.com/> (bez udania dátumu)
- [3] 7.5_Profit_Analyzer_User_Guide.pdf, <http://www.vendavo.com/> (bez udania dátumu)
- [4] 7.5_Price_Manager_Implementation_Guide.pdf, <http://www.vendavo.com/> (bez udania dátumu)
- [5] Technical Training Guide: Vendavo Basics 1, <http://www.vendavo.com/> (bez udania dátumu)
- [6] Technical Training Guide: Vendavo Basics 2, <http://www.vendavo.com/> (bez udania dátumu)
- [7] Blog Ing. Milana Borůvku, <http://blog.boruvka.net/> (bez udania dátumu)
- [8] Webové stránky firmy JetMinds s.r.o., <http://www.jetminds.com/> (bez udania dátumu)